

# MODERN C++ - PRE-TEST

Name: .....

Date: ..... Score: .....

Question	1	2	3	4	5	6	7	8
Answers								

This is a **multiple choice** test. More than one answer may be correct.

Assume **C++14** for following questions.

Assume that all necessary includes are present.

Good luck!

1. Suppose we have following methods defined:

```
void foo(int);
```

```
void foo(void*);
```

```
void foo(std::nullptr_t);
```

Which method will be called in this case:

```
foo(nullptr);
```

a) void foo(int);

b) void foo(void\*);

c) void foo(std::nullptr\_t);

d) ambiguous call

2. What is the type of variable v?

```
const int i = 42;
```

```
auto & v = i;
```

a) int

b) const int&

c) const int\*

d) int&

e) none of above answers is valid

3. Which of the following are the correct ways to iterate over a collection?

```
std::vector<int> v;
```

a) for(std::vector<int>::size\_t i = 0; i < v.size(); ++i)

b) for(auto it = v.begin(); it != v.end(); ++it)

c) for(auto & item : v)

d) std::for\_each(std::begin(v), std::end(v), [](auto & item){});

4. What is the type of variable d?

```
int array[12];
```

```
auto d = array;
```

a) const int\*

b) int\*

c) int(&)[12]

5. Which of the following elements can be defined as deleted (= delete):

- a) default constructor
- b) copy constructor
- c) move constructor
- d) copy assignment operator
- e) move assignment operator
- f) destructor
- g) free function
- h) class method

6. Which lambda function is valid?

- a) `[]() -> int { return 4; };`
- b) `int [](){ return 4; };`
- c) `auto [](){ return 4; };`
- d) `[]() -> auto {return 4; };`
- e) `[](){ return 4; };`
- f) `[] { return 4; }`
- g) `[] -> int { return 4; }`
- h) `int []{ return 4; }`

7. What kind of problem do we have here?

```
#include <memory>
struct Gadget {
    void call_method() { cout << "hi"; }
};
void sink(std::unique_ptr<Gadget> gdgt) {
    gdgt->call_method();
}
int main () {
    auto ptr = std::make_unique<Gadget>();
    sink(std::move(ptr));
    ptr->call_method();
    return 0;
}
```

- a) Memory leak
- b) Undefined behavior
- c) Compiler error
- d) Linker error

8. Which overrides are syntactically valid?

- a) `void doSth() const override;`
- b) `void doSth() const = override;`
- c) `void doSth() override const;`
- d) `virtual void doSth() const override;`
- e) `virtual void doSth() = override const;`